AFRL-RH-WP-TR-2020-0119

LEARNING TO AUTOMATE SOCIAL ENGINEERING RESISTANCE (LASER)

Dawn Song
Univ. of California, Berkeley
2150 Shattuck Ave, Rm 313
Berkeley, CA 94704

Ben Y. Zhao
Univ. of Chicago
5801 S. Ellis Ave
Chicago, IL 60647

Bo Li
Univ. of IL at Urbana-Champaign
901 West Illinois Street
Urbana, IL 61801

Le Song
Georgia Institute of Technology
North Ave NW
Atlanta, GA 30332

Chris Re
Percy Liang
Stanford Univ
450 Serra Mall
Stanford, CA 94305

June 2020

Final Report

AIR FORCE RESEARCH LABORATORY
711th HUMAN PERFORMANCE WING
AIRMAN SYSTEMS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

Qualified requestors may obtain copies of this report from the Defense Technical Information Center (DTIC).

AFRL-RH-WP-TR-2020-0119 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//signature//
ERIC HANSEN, DR-III
Mission Analytics Branch
Warfighter Interactions and Readiness Division
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

//signature//
WILLIAM P. MURDOCK, DR-IV, Ph.D.
Chief, Mission Analytics Branch
Warfighter Interactions and Readiness Division
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

//signature//
LOUISE A. CARTER, DR-IV, Ph.D.
Chief, Warfighter Interactions and Readiness Division
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
30-06-2020	Final	20 September 2018 – 29 March 2020

4. TITLE AND SUBTITLE

Learning to Automate Social Engineering Resistance (LASER)

5a. CONTRACT NUMBER
FA8650-18-C-7882

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dawn Song (Univ of CA-Berkeley) Le Song (Georgia Institute of Tech)
Ben Y Zhao(Univ of Chicago) Chris Re (Stanford Univ)
Bo Li (Univ of IL at UC*)* Percy Liang (Standford Univ)

5d. PROJECT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)

Univ of CA – Berkeley, 2150 Shattuck Ave, Rm 313, Berkeley, CA 94704
Univ of Chicago, 5801 S. Ellis Ave, Chicago, IL 60647
Univ of IL at UC; 301 West Illinois Street, Urbana, IL 61801
Georgia Institute of Technology, North Ave NW, Atlanta, GA 30332
Stanford Univ; 450 Serra Mall; Stanford, CA 94305

5e. TASK NUMBER

5f. WORK UNIT NUMBER
H0X6

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
711th Human Performance Wing
Airman Systems Directorate
Wright-Patterson AFB, OH 45433

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RH-WP-TR-2020-0119

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution A. Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
AFRL-2021-0470; Cleared 19 Feb 2021

14. ABSTRACT

Project LASER explores and creates a suite of technologies that can radically harden enterprise security by the large scale automation of active social engineering defenses. The proposed LASER system is able to efficiently process large volume of messages, accurately detecting the attacks and the attack motives, and actively responding back to the attacker to obtain more information through well-formed high-quality response. The proposed work will shield both individuals and organization from social engineering attacks.

15. SUBJECT TERMS
ASED

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Eric Hansen
Unclassified	Unclassified	Unclassified	SAR	23	19b. TELEPHONE NUMBER *(include area code)*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The Learning to Automate Social Engineering Resistance (LASER) team thanks the Purdue team for collaborating on the technical area (TA) 1 classifier component design. The LASER team thanks the Carnegie Mellon University (CMU) team for collaborating on the TA2 dialogue system component design. The LASER team also thanks Jet Propulsion Laboratory (JPL) for their infrastructure and evaluation support.

1.0 SUMMARY

Social engineering is a subtle, highly effective attack strategy where an attacker misleads and manipulates humans to achieve a desired end, e.g., to ask users to wire money to the attacker's account or to ask users to provide their account credentials. Far more insidious than simple email spam, the technique has been used to steal millions of dollars from high-profile tech companies like Google and Facebook. In fact, social engineering is already the most common way in which computer systems are initially breached. The prevalence and sophistication of such attacks can only be expected to increase as machine learning frameworks make powerful models for knowledge discovery and interactive dialogue more accessible to end users (and thus to attackers), making it hard, if not impossible, for end users to detect such attacks and prevent damages from happening. Accordingly, there is an urgent need for artificial intelligence (AI) and security researchers to proactively create intelligent systems to mitigate such threats posed by sophisticated attackers and the misuse of other AI systems.

In this project, we (University of California Berkeley (UC Berkeley), University of Illinois at Urbana-Champaign (UIUC), University of Chicago (UChicago), Stanford University, Georgia Institute of Technology (GaTech) proposed the design and development of an automated attack detection and attacker identification system. We dub this system LASER. The LASER system combines techniques from machine learning (ML), security, natural language processing (NLP), data/text mining, and knowledge graphs, and is designed to address challenges in two technical areas: attack detection using passive and active detection engines (TA1), and proactive adversary engagement and identification (TA2). Responding to both TA1 and TA2, LASER performs both passive detection and active detection from multiple dimensions, as well as a scalable, highly available, and fault tolerant system architecture that is specially designed for active social engineering defense (ASED). LASER is able to respond to heterogeneous types of social engineering attacks, including emails, LinkedIn messages, and short message services (SMS). Specifically, for TA1 passive attack detection, LASER collects evidences from multiple dimensions such as message body and meta data, and uses a combination of ML (XGBoost) and deep learning (DL) (DistillBERT, RoBERTa) approaches to determine if an incoming message is a friend or a foe, as well as its motive category. Furthermore, LASER extracts useful entities and relations from the message body and stores the information in a local knowledge base, in order to have a better understanding of the incoming message and facilitate later response generation tasks. For TA2 active attack detection, LASER (by collaborating with the CMU team) adopts a combination of template-based and learning-based approaches to generate proper and natural responses based on the extracted information to facilitate the dialogue conversion. Furthermore, LASER provides a fake document cloud component which acts as a honey pot to further collect sensitive information from a possible attacker, as well as an unanswerable question generation component to determine whether the other end is a human or a bot. For the system design and architecture, LASER adopts a micro-service design style to support hot upgrade. LASER is also stateful, so that it can resume to process the remaining messages after the accidental shut down. Specifically, for the TA1 classification, LASER has a plugin style classification framework so that different types of extractors and models can be easily declared and integrated in the system, and the dependency flow between components can be automatically computed. The system architecture has several major optimizations so that it is highly reliable and available, and can achieve high throughput in processing a large volume of incoming messages. Overall, LASER enables automatic detection and active investigation of a wide range of social engineering attacks, benefiting both individuals and organizations.

Throughout the one and a half years of the ASED program, the LASER team has participated in all engagements, workshops, dry-runs, and two major system evaluations organized by Defense Advanced Research Projects Agency (DARPA). The proposed LASER system has demonstrated its performance in handling high volumes of social engineering attack messages, processing heterogeneous types of social engineering attacks, and achieving reasonable results. Papers that were supported by the ASED funding were also published in premier cybersecurity and AI conferences, including Institute of Electrical and Electronics Engineers (IEEE) Security and Privacy (S&P), Computer and Communications Security (CCS), Network and Distributed System Security (NDSS), International Conference on Learning Representations (ICLR), Empirical Methods in Natural Language Processing (EMNLP), International Joint Conference on Artificial Intelligence (IJCAI), and International Conference on Machine Learning (ICML).

The LASER project is a collaboration among UC Berkeley, UIUC, UChicago, Stanford,and GaTech. UC Berkeley has acted as the prime contractor while other organizations were sub-contractors to UC Berkeley. For ASED evaluations, the LASER team partnered with the Purdue team on TA1 classifier design and the CMU team on TA2 classifier design. This report only contains the work done by the LASER team led by UC Berkeley.

2.0 LASER SYSTEM DESIGN

2.1 Overall System Architecture

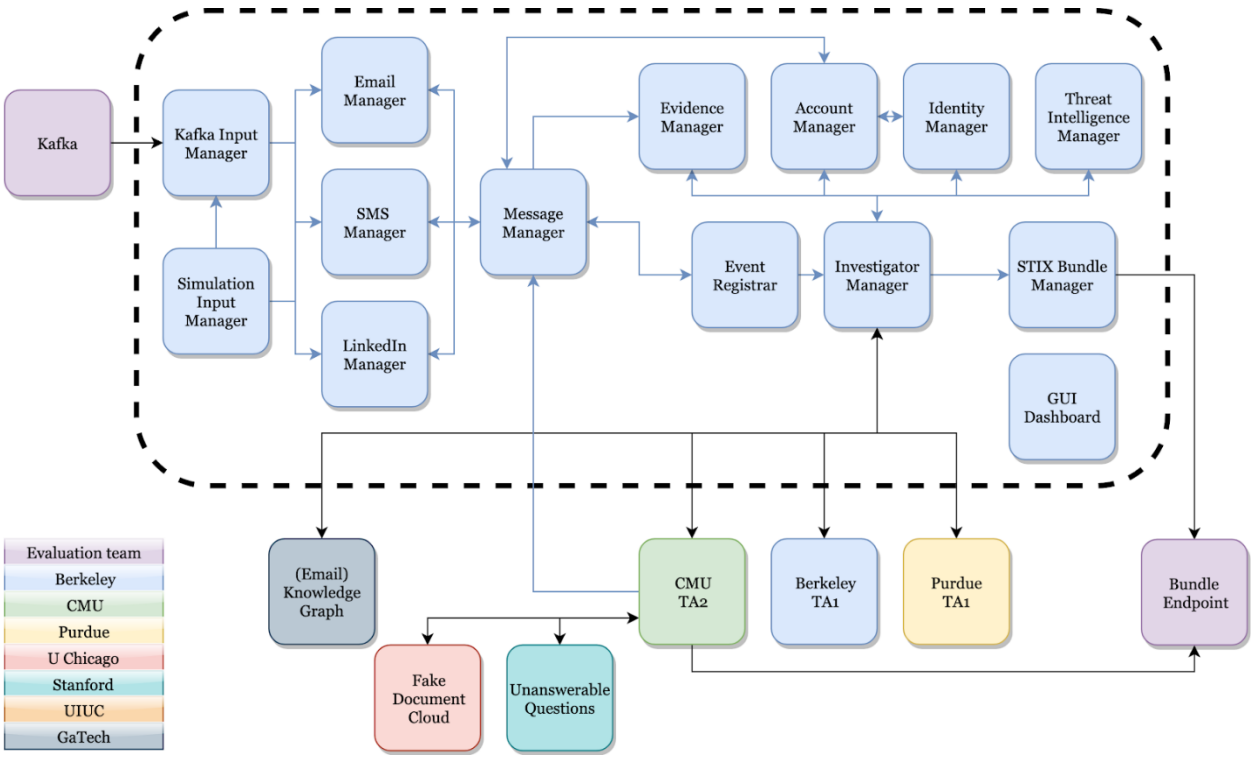

Figure 1. The LASER System Architecture.

Figure 1 shows the overall architecture of LASER. Each square in the figure represents a component in the system. Each component is an individual containerized web server that supports a specific set of Application Programming Interfaces (API). Some components can also actively reach out to collect information and send out bundles (e.g., CMU TA2 component). If the component needs access to Internet, databases or other components, access behaviors are failure-tolerant and the endpoints (e.g., website/database Uniform Resource locator [URL]) are configurable in the docker environment variables.

2.1.1 Input

There are three categories of input to LASER.

- Kafka: All messages will be pushed to kafka and picked up by our Kafka input manager. They will be processed through the pipeline and saved in the databases.
- Direct inbox access: This functionality means if we have direct access to the (email) inbox, we can directly connect to them and scan their content.
- Internet services: Some of the component may need Internet access to achieve better performance or use extra functions (e.g., CMU TA2 component may need access to fake document cloud component to get information of attackers).

2.1.2 Output

There are two categories of output of LASER.

- Evaluation Endpoint: Structured Threat Information Expression (STIX) bundles will be submitted to reflect the judgement of our system. For TA1, all classifiers will first submit their results to the investigation manager and the combined classification result (through voting) will be sent to the endpoint. For TA2 dialogue system, CMU TA2 component will handle this part itself.
- Graphical User Interface (GUI) dashboard: All the components in LASER support automatic status reporting through a frontend GUI to help run diagnosis.

2.1.3 Message Access & Storage

We provide message access in two styles.

- Passive: The component will receive an API call whenever a new message comes into the system. A response that contains the judgement within several seconds is expected.
- Active: The system will provide API for component to query the entire message history and the detailed information of each message. The component itself is responsible for submitting their results to both the evaluation endpoint and our system.

2.1.4 Unit Test & Internal Dry Run Support

Skeleton code (including Software Development Kit [SDK]) and container-level unit tests are provided to check the correctness of the component. For all types of API requirements, we designed dummy cases and tested the efficiency of the responses (e.g., classifier should respond to the request within three seconds). For the active components and the components that require outbound access, we also provided dummy service during the test. Once all (or a necessary set of) components passed the unit tests, they were integrated in the system and the system was deployed on the Kubernetes infrastructure. An internal dry run was also conducted to reduce the possibility of failure during the official ASED evaluation.

2.1.5 Summary & Highlights

In summary, the LASER system supports the processing of heterogeneous social engineering sources (emails, LinkedIn messages, SMS), and it modular and extensible. Following are key highlights of the system.

- Unified representation of heterogeneous social engineering attack messages
- Unified representation of account-level/identity-level relationships in form of graphs
- Better storage model: efficient access to message thread and communication history
- Service-based model that support hot upgrade and horizontal scaling
- Stateful design with the database porter which maintains the state of the system so that it can resume to process the remaining messages after the accidental shut down. Separate records of all submitted bundles are also maintained.

- Separation between features and classification method, a plugin-style classification framework that supports the easy integration of multiple feature extractors and classification method and further enable AutoML techniques
- Delayed task and regularly recurring task are supported.

Overall, the system is highly scalable and available, as can be demonstrated from the results of DARPA ASED official evaluation.

2.2 LASER SYSTEM User Interface (UI)

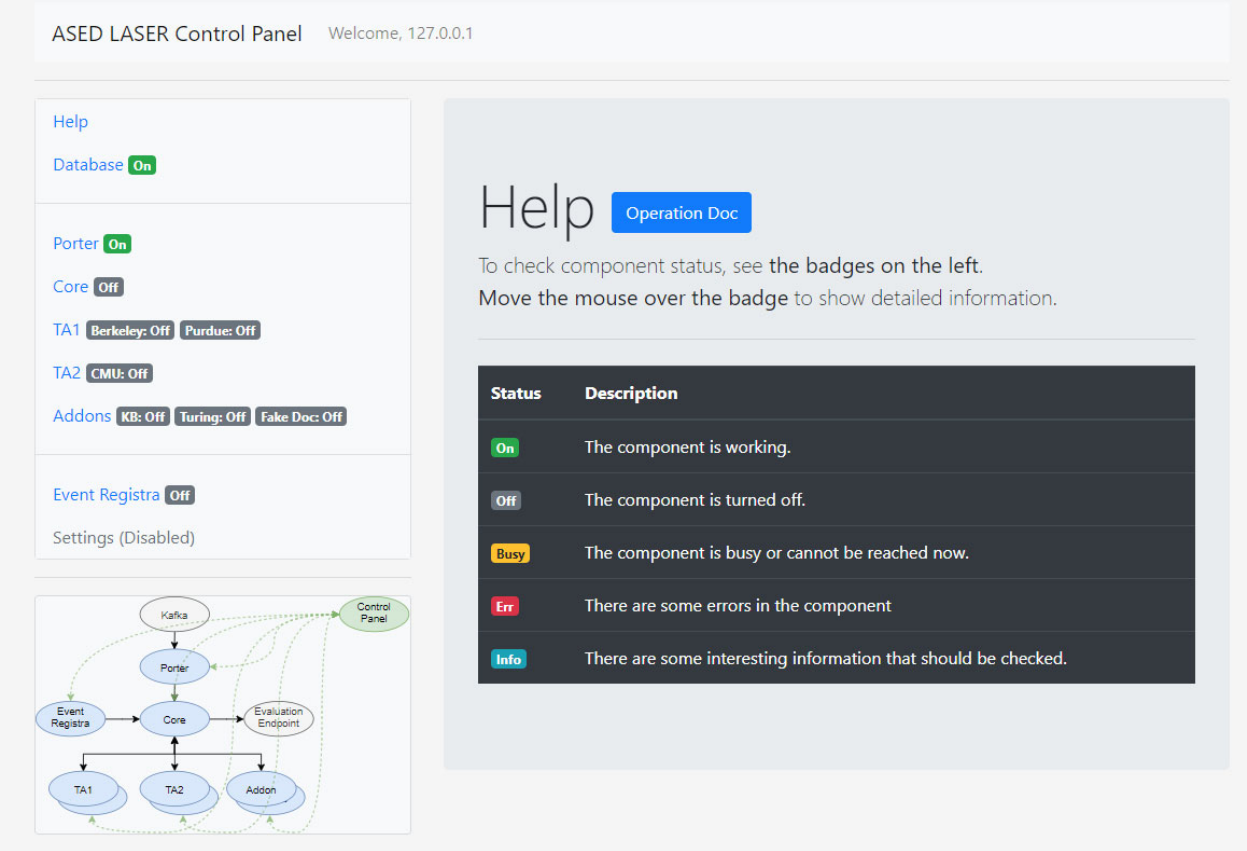

Figure 2. The LASER System UI.

To ease the easy monitoring of system component status and diagnose possible failures, LASER system comes with an informative UI. As shown in Figure 2, the user can interact with the UI control panel to view the status of different components. The UI is specially designed to have all sensitive Personally Identifiable Information (PII) removed.

The status of the micro-services is labeled on the left-hand side. During the evaluation, this feature is used to confirm whether our deployment is successful and will reflect the input status changes from the evaluation team. There is also more detailed access to database status (statistics for different message types), sampled messages (only hash value, PII removed) and classifier log, which can be accessed by clicking on those tabs. For example, in the porter tab, we can access the status of Kafka queue and see how many messages are actually loaded to our system.

To check the aliveness for external services like fake document cloud, we also maintain their status on our dashboard. By observing the dashboard, it's easy to tell whether the system is working as expected.

2.3 TA1 Component

2.3.1 TA1 Classifiers

LASER adopts a hybrid approach that collects evidence of an incoming message (email, LinkedIn message, SMS) from multiple dimensions, trains different ML and DL models, and vote the final classification result based on the results of individual models. In the latest version of LASER, three classifiers are adopted:

- XGBoost metadata classifier: We extract features from the metadata such as headers and their value, message mime type sequence, keyword existence in the eml file, and train a XGBoost classifier to make the classification.
- DistilBERT semantic classifier: We remove all metadata and structural information from the email message and train a DistilBERT model.
- RoBERTa classifier with adversarial training: We leverage state-of-the-art adversarial training techniques on a RoBERTa classifier to help it generalize to unknown examples.

Note that the XGBoost classifier and the RoBERTa classifier are only used for classifying incoming emails, as they require additional information from the eml file such as meta data. The DistilBERT is trained only using the message body part of emails, and is used to make predictions for all three types of incoming messages.

2.3.1.1 Details on Classifier Development

We present the details on how we train the three types of classifiers. We use data from five sources (apwg, JPL Abuse 2017, JPL historic, Classified Volunteer Emails and Enron) to train our model and evaluate the model on both the original dataset and the dry-run data. The detailed characteristics and statistics of these different sources are discussed in the internal evaluation section in Section 3.1.

2.3.1.2 Classifier Ensembling

Once we trained all classifiers, for an incoming message, the TA1 component of LASER aggregates individual prediction results to make a final prediction. The current scheme for ensembling is through majority voting.

During ASED official evaluation, as our TA1 component is combined with Purdue TA1 component, our classification result was further aggregated with the result of Purdue's component to make a final decision on the label of an incoming message.

2.3.2 Adversarial Training

We further provide some background on the details of our RoBERTa, which we leveraged adversarial training.

The main challenge of passive detection is that the data distribution of the phishing email and the general email is sparse. That is, the number of phishing email is very small compared to the number of general benign emails, and it might be difficult to know the exact distribution of an

enterprise and obtain enough training data that meets such distribution accurately. If the model is only trained using the training datasets provided by ASED (details in Section 2.3.1), the model might not be able to effectively identify the labels of unknown data distribution. As a result, high false alarm rate may occur when evaluating the system against unseen phishing emails. Therefore, we need to improve the generalization of the model to unknown distributions.

Recent work has proposed to use adversarial training to improve the generalization of language understanding models [11]. The basic principle of adversarial training is to construct the adversarial samples by adding perturbations to the original samples to improve the robustness of the model when it encounters the adversarial samples. At the same time, it can also improve the model's performance and generalization ability to a certain extent.

The ICLR)2018 paper titled “Towards DL Models Resistant to Adversarial Attacks” proposes a technique called Projected Gradient Descent (PGD) [11]. The PGD method is widely regarded as the most effective method to defend against adversarial samples. It trains model parameters by constructing adversarial samples through several steps of inverse gradient descent in the epsilon range. However, PGD's k-step adversarial sample generation needs to increase the original network training time by k times, which is difficult to scale up to large-scale data. A more advanced model (“FreeLB: Enhanced Adversarial Training for Natural Language Understanding” in ICLR 2020) proposes further optimizations of PGD [12]. By accumulating gradients, the time consumption of multiple gradient descents is avoided. This makes it take almost no extra time to generate k-step adversarial samples.

In the current version of the LASER system, we have implemented PGD-based text representation learning adversarial training and applied it to the RoBERTa model. On our local training / verification set of passive detection, the accuracy of the original RoBERTa is 91.4%, and 91.7% after using PGD. Note that the training / validation set for this dataset is from the same distribution. When dealing with real data with unknown distribution, such as the high false alarm rate caused by unknown distribution, it can be expected that this method will have a more significant improvement.

In terms of other ways to improve the generalization capability of the model, multi-task learning and transfer learning are two typical methods that can be considered. However, due to the large difference in data distribution between different email datasets, according to the past study, multi-task learning or transfer learning can easily lead to negative transfer. Therefore, in LASER, we leveraged adversarial training to improve the generalization capability of the model.

2.3.3 Email Knowledge Base

We constructed an email knowledge base that contains the extracted useful information from a collection of email message bodies. In this project, we leveraged Enron email dataset to develop our automated knowledge extraction tool. Our tool was built on top of Spacy [13] and Open Information Extraction (IE) [14]. The extracted knowledge from emails can be roughly categorized as a set of entities (e.g., person name), entity properties, and entity relations. For entity recognition, we used Spacy Part of Speech (POS) tagging to extract nouns. For property extraction, we first used a set of heuristics to preprocess the emails to only retain “entity words.” We then used Latent Dirichlet Allocation (LDA) to compute a set of topics from the given email dataset in the form of word-topic matric. Finally, we assigned the entities to the topics with largest probabilities. For relation extraction, we used Open IE tools to extract the binary relations. The constructed email knowledge graph structuralizes the information in the

unstructured email corpora, which is useful to empower a wide range of downstream applications such as question & answer (QA), dialogue generation, and spam detection.

2.4 TA2 Component

2.4.1 Dialogue System

The dialogue system component was mainly developed by the CMU team by calling the SDK provided by our LASER system. The component leverages a combination of template-based and learning-based approaches to generate proper response of an incoming message. In the template-based approach, the system will extract information from each turn of the input of the attacker into a structured representation. This representation is used to update the dialogue state. Given a dialogue state, the dialogue policy module will conduct actions which will be later rendered into natural language sentences by filling relevant information into templates prepared in advance. Learning techniques (like disfluency detection) are used to make the output more natural. By continuously interacting with the potential attacker using this procedure, we are actively gathering information from the attacker as well as wasting their effort.

Besides system infrastructure support, our LASER team further proposed the design and development of two components to facilitate the active attacker identification, which we will present next.

2.4.2 Fake Document Cloud

We designed a fake document cloud component which serves as a honey pot to trick the attacker and extract further sensitive information. When a particular pattern occurs in an incoming email, a link to the fake document cloud service is generated and embedded in the natural language response to the incoming message. When the sender clicks the link, he/she will access our hosted fake document cloud website, which mimics a genuine financial service website. Our fake document cloud website uses various techniques to extract information from attackers, including Internet Protocol (IP) address, user agent, language, browser fingerprint. Tor/Virtual Private Network (VPN) users are "blocked for security reasons, please disable." The attacker needs to create an account, fill in the form (name, location, etc.) to get what they thought should be in the honeypot.

2.4.3 Unanswerable Question Generation

We designed an automated Turing test engine that could identify if human or an autonomous bot sent the email. From the development set for the SQUAD2.0 dataset [15], which focuses on automatic QA, we isolated 200 examples on which 111 models had very low average overlap with the correct answer. We used these examples to test if the attacker is a bot based on the answer that was selected.

For example, current QA models perform poorly on questions such as "in what country is Normandy located?" given a paragraph of text description about the Normans. If the attacker can successfully answer these questions, at least we can tell the reply is not an automated procedure. In the context of social engineering attack, questions related to knowledge about owner of the email address can be generated and included as a part of the email. And to prove the message is not from a bot, the potential attack needs to reply with answers that contains specific knowledge about their target (such as relations between the victim and a third person).

3.0 RESULTS AND DISCUSSION

3.1 Internal Evaluation Results

3.1.1 Dataset

Table 1. Dataset Statistics – Number of Emails

Source	#instance
APWG	1,502,372
JPL Abuse 2017	12,209
JPL historic	3,245
Volunteer Emails	7,591
Enron	517,401

We use data from five different sources to train our models to make it robust (see Table 1). For each source, we split the instance into training set (80% instances), dev set (10% instances), and test set (10% instances). After the 19 Fall dry-run, we also include the dry-run data as part of the dev set and consider it as one of our metrics.

Table 2. Dataset Statistics – Number of Instances for each Motive

Source	#instance
malicious	1,476,473
benign	528,679
acquire_credentials	1,320
acquire_pii	178
annoy_recipient	1,559
build_trust	1,190
install_malware	3,053

In addition, to support motive classification, we also tag all the instance with their motive labels. Considering that one email could contain multiple labels, we formulate the task as multiple binary classification problems and build pretrained embedding based model for it.

3.1.2 Evaluation Results

Table 3. Performance of Biodirectional Encoder Representations from Transformers (BERT)-Based Model.

	acc	precision	recall	f1
predict_if_malicious	94.03	91.03	97.68	94.24
predict_if_acquire_pii	99.83	100.00	5.56	10.53
predict_if_annoy_recipient	98.67	56.53	63.78	59.94
predict_if_build_trust	98.94	72.41	17.65	28.38
predict_if_install_malware	97.94	70.54	55.65	62.21

Considering the meta-data of emails from the test environment might be different from our observation, we first test our BERT-based model for text classification on the test set mentioned above. As we can see in the Table 3, our model performs very well on the "friend or foe" classification. For motive prediction, although the size of training instances is relatively small, the model still managed to provide reasonable performance.

3.2 ASED Fall 2019 Official Evalution Results

The LASER team participated in the ASED Fall 2019 Official Evaluation in August 2019. In this section, we present the evaluation results and discussion.

For TA1 classification, the LASER team led by UC Berkeley, partnered with the Purdue team: each team proposed a classifier, and the final prediction result was determined by the final aggregated score (average of the scores of the two classifiers) compared with the 0.5 threshold. In this evaluation, the BERT classifier trained using the message body of emails was submitted by our team.

3.2.1 Evaluation Results

Here are the evaluation results. In total, the LASER system received 2,034 total messages, which is the number of messages in the message ledger. Among them, there were 78 attack messages in total, and 50 of them were successfully identified as foe by the LASER system. For friendly messages, there were 1,804 messages whose STIX bundles matched the message IDs in the ledger. Among them, 394 were identified as foe by the LASER system. In summary, the recall of the aggregated TA1 classifier was 50/78 = 64%, and the false alarm rate was 394/1804 = 22%.

3.2.2 Discussion

During the evaluation, there were some glitches and incorrect operations by of our system by the evaluation team (i.e., JPL), which made our system only respond to a subset of attack messages that we received. As demonstrated by the k8s running status report, our system skipped almost all emails before the last week of evaluation due to an incomplete restart. Although we provided necessary documentation and scripts to restart the system correctly from the first email and rerun all classification logics, there was a misunderstanding in operations of the evaluation team that led to an incomplete restart. This eventually affected the recall metric of our TA1 classifier.

Besides, the classifier proposed by our team was integrated with the classifier from the Purdue team. Due to the issues in format compatibility of submitted bundles, some of the final STIX

bundles were rejected by the format checking procedure, which also affected the metrics of our combined classifier.

Furthermore, there was a significant distribution gap between the dataset we used to develop the classifier and the attack emails provided by Thomson Reuters Special Services (TRSS) to evaluate the classifier. As demonstrated in our internal evaluation results in Section 3.1, our classifier could achieve more than 90% accuracy and more than 90% F1 score, which were both significantly higher than the metrics we achieved in the fall evaluation.

3.3. ASED Winter 2020 Official Evaluation Results

The LASER team participated in the ASED Winter 2020 Official Evaluation in January 2020. In this section, we present the evaluation results and discussion.

For TA1 classification, the LASER team led by UC Berkeley partnered with the Purdue team. As described in Section 2.3.1, our team proposed three classifiers: XGBoost, DistillBERT, and RoBERTa, and our classifiers were combined with Purdue classifier via majority voting.

3.3.1 Evaluation Results Released by JPL

Here are the evaluation results on the official metrics slides released by JPL. In total, the LASER system received 50785 email messages. Among them, there were 524 total foe messages (True Positive [TP]+False Negative [FN]), and 5,0261 total friendly messages (False Positive [FP]+True Negative [TN]). The LASER system made 11,167 total foe predictions (TP+FP), and 221 of them were correct foe predictions (TP). The LASER system made 10,946 incorrect foe predictions (FP). For final metrics, the combined TA1 classifier achieved 2% precision, 42% recall, 22% false alarm rate, and 0.04 F1 score.

For LinkedIn messages, the LASER system received 5,389 total messages, and all of them were total foe messages (TP+FN). The total foe predictions of the LASER system were 16, and all of them were correct foe predictions. Thus, the system achieved 100% precision, 0% recall, and 0.01 F1.

For SMS messages, the LASER system received 163 total messages, and all of them were total foe messages (TP+FN). The total foe predictions of the LASER system were 39, and all of them were correct foe predictions. Thus, the system achieved 100% precision, 24% recall, and 0.39 F1.

3.3.2 Discussion

It is important to note that the metrics described in the previous section were from the combined classifier, which we partnered with the Purdue team. We next present the detailed analysis of the classifier proposed by our team led by UC Berkeley (i.e., Berkeley classifier in short).

For emails, the metrics slides show that LASER has 42% recall (called “accuracy” in the last evaluation) and 22% false alarm rate (FAR). We would like to note that this is for the combined classifier. For our Berkeley classifier, we achieved 64.9% recall and 9.6% FAR. The reason for the performance degradation for the combined classifier is because we did voting with Purdue classifier in a non-ideal way.

Furthermore, for emails, our recall calculation was largely affected by a possible infrastructure breakdown on 02/25. In total, we received 524 attack emails, including 420 initial emails and 104 route back emails. During that breakdown period, the STIX bundle submissions for

incoming route back messages of all teams failed. However, given that we have the largest number of route back messages due to our prolific TA2 component (e.g., 104 messages for us v.s. 4 messages for PIRANHA team), our team was affected the most. In particular, there are 76 route back messages for which our bundle submissions were rejected. Furthermore, including those route back messages in the recall calculation does not seem to be a very fair baseline to us, as teams that receive more such messages will have a larger numerator (e.g., 524 for us, 424 for PIRANHA). If we only consider the initial 420 attack messages that are the same across all teams, the recall of our Berkeley classifier then improves from 64.9% to 81.0%.

For SMS, the combined classifier achieved 24% recall, which is the highest among all teams. This result came from our Berkeley classifier, particularly the DistillBERT sub-classifier (Purdue classifier didn't make predictions for SMS and LinkedIn). Furthermore, since we have no training data for SMS, our DistillBERT sub-classifier was trained on the message body of emails only. Such results demonstrate that our deep learning-based approaches have certain capability to generalize to other unseen domains, and we can definitely do better if we have training data for SMS for the next round.

We admit that, due to the lack of training data (both friend and foe) that resembles the evaluation data distribution, our false alarm rate is not ideal. To mitigate the issue, we have been exploring the use of adversarial training and transfer learning techniques to improve the model generalization and robustness. Our RoBERTa classifier with PGD -based adversarial training has demonstrated certain capability in this evaluation, and we will explore more along this direction.

The evaluation results demonstrate that the LASER system architecture that our Berkeley team has built is highly reliable and available, and achieved high throughput. Our system can easily support hot upgrades, Continuous Integration (CI)/ Continuous Delivery (CD), and horizontal scaling. During the evaluation, CMU TA2 component has requested the upgrade multiple times with more computation resources, and the restart of these components does not affect the running status of other parts of the system. Our system adopts a micro-service design style and is stateful, so that it can resume to process the remaining messages after the accidental shut down. Furthermore, we adopt a plugin style design to support TA1 functionalities, so that multiple feature extractors and different classifiers can be easily implemented and integrated via policy specifications.

4.0 RESEARCH OUTCOMES

During the course of this work, besides the design and development of the LASER system and relevant techniques, the LASER team has also published a collection of papers in premier venues in cybersecurity and AI, including IEEE S&P, CCS, NDSS, ICLR, EMNLP, IJCAI, and ICML [1-10].

5.0 CONCLUSIONS

The LASER team has researched and developed a suite of technologies to harden enterprise security against social engineering attacks. We proposed the design and development of an automated attack detection and attacker identification system that answers both TA1 and TA2 responsibilities. The team has participated in all engagements and evaluations in Phase I throughout the program with success. Through DARPA ASED evaluations, we have demonstrated the practical efficacy of our system in scaling to large volume of data, accurately predicting labels for incoming messages, and generating high-quality natural language responses.

6.0 REFERENCES

[1] “Et Tu Alexa? When Commodity WiFi Devices Turn into Adversarial Motion Sensors,” Yanzi Zhu, Zhujun Xiao, Yuxin Chen, Zhijing Li, Max Liu, Ben Y. Zhao, Haitao Zheng; NDSS 2020

[2] Learn to Explain Efficiently via Neural Logic Inductive Learning Yuan Yang, Le Song;ICLR 2020

[3] Latent Backdoor Attacks on Deep Neural Networks; Yuanshun Yao, Huiying Li, Haitao Zheng, Ben Y. Zhao; CCS 2019

[4] Certified Robustness to Adversarial Word Substitutions; Robin Jia, Aditi Raghunathan, Kerem Göksel, Percy Liang; EMNLP 2019

[5] Performing Co-Membership Attacks Against Deep Generative Models; Kin Sum Liu, Chaowei Xiao, Bo Li, and Jie Gao; ICDM 2019

[6] Efficient Task-Specific Data Valuation for Nearest Neighbor Algorithms, Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas Spanos, Dawn Song; VLDB 2019

[7] Robustra: Training Provable Robust Neural Networks over Reference Adversarial Space, Linyi Li, Zexuan Zhong, Tao Xie, Bo Li; IJCAI 2019

[8] Scaling Deep Learning Models for Spectrum Anomaly Detection; Zhijing Li, Zhujun Xiao, Bolun Wang, Ben. Y. Zhao, Haitao Zheng; MobiHoc 2019

[9] Robust Inference via Generative Classifiers for Handling Noisy Labels; Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, Jinwoo Shin; ICML 2019

[10] Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks; Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, Ben Y. Zhao; IEEE S&P 2019

[11] Towards Deep Learning Models Resistant to Adversarial Attacks; Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu; ICLR 2018

[12] FreeLB: Enhanced Adversarial Training for Natural Language Understanding; Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, Jingjing Liu; ICLR 2020

[13] Spacy: Industrial-Strength Natural Language Processing, https://spacy.io/

[14] Open IE: Stanford Open Information Extraction, https://nlp.stanford.edu/software/openie.html

[15] Know What You Don't Know: Unanswerable Questions for SQuAD; Pranav Rajpurkar, Robin Jia, Percy Liang; ACL 2018

7.0 LIST OF ACRONYMS, ABBREVIATIONS AND SYMBOLS

AI	Artificial Intelligence
API	Application Programming Interface
ASED	Active Social Engineering Defense
BERT	Biodirectional Encoder Representations from Transformers
CCS	Computer and Communications Security
CD	Continuous Delivery
CI	Continuous integration
CMU	Carnegie Mellon University
DARPA	Defense Advanced Research Projects Agency
DL	Deep Learning
EMNLP	Empirical Methods in Natural Language Processing
FAR	False Alarm Rate
FN	False Negative
FP	False Positive
GaTech	Georgia Institute of Technology
GUI	Graphical User Interface
ICLR	International Conference on Learning
ICLR	International Conference on Learning Representations
ICML	International Conference on Machine Learning
IEEE	Institute of Electrical and Electronics Engineers
IJCAI	International Joint Conference on Artificial Intelligence
IP	Internet Protocol
JPL	Jet Propulsion Laboratory
LASER	Learning to Automate Social Engineering Resistance
LDA	Latent Dirichlet Allocation
ML	Machine Learning
NDSS	Network and Distributed System Security
NLP	Natural Language Processing
Open IE	Open Information Extraction
PGD	Projected Gradient Descent
PII	Personally Identifiable Information
POS	Part of Speech

QA	
S&P	Security and Privacy
SDK	Software Development Kit
SMS	Short Message Service
STIX	Structured Threat Information Expression
TA	Technical Area
TN	True Negative
TP	True Positive
TRSS	Thomson Reuters Special Services
UC Berkeley	University of California Berkeley
UChicago	University of Chicago
UI	User Interface
UIUC	University of Illinois at Urbana-Champaign
URL	Uniform Resource Locator
VPN	Virtual Private Network